

Umbral de problemas en ingeniería de sistemas y programas afines: una visión desde lo cotidiano

Problem threshold in systems engineering and related programs: a vision from the everyday life

Ricardo Botero Tabares

Candidato a Magíster en Ingeniería (Área Sistemas y Computación)
Especialista en Didáctica Universitaria
Ingeniero de Sistemas
Profesor Asociado Facultad de Informática
Tecnológico de Antioquia – Institución Universitaria
rbotero@tdea.edu.co

Resumen

Este artículo, producto de una revisión de tema, narra el origen de algunos problemas típicos en ingeniería de sistemas, ingeniería de software y programas afines, causados por situaciones cotidianas que con el transcurrir del tiempo, se convierten en modelos de estudio en el proceso formativo de los noveles ingenieros.

Palabras clave: problemas típicos en ingeniería, estudio de la ingeniería.

Abstract

This article, resulting from a subject review, traces back the origin of some typical problems caused by everyday situations in systems engineering, software engineering and related programs, those programs become as time goes by, models to study in the junior engineer learning process.

Keywords: typical problems in engineering, engineering study.

Introducción

Estudiar programas académicos relacionados con ciencias de la computación o áreas afines conlleva el encuentro con currículos que incluyen asignaturas como lógica de programación, estructura de datos, sistemas operativos, análisis y diseño orientado a objetos, teoría de redes, entre otras, en las cuales ciertos temas de tinte estrictamente científico se asemejan a hechos reales, en ocasiones rutinarios, hogareños, ciudadanos o idiosincrásicos, que generan situaciones problemáticas solucionadas en gran parte mediante el uso de los modelos matemáticos inherentes a la investigación de operaciones, las matemáticas discretas, la geometría analítica, la física o el cálculo diferencial e integral. Esto convierte al estudio, momentáneamente, en un juego capcioso que requiere del ingenio de las personas en búsqueda de una solución óptima, haciendo de lo cotidiano una eclosión de descubrimientos que pueden verse en nuevos desarrollos científicos.

Son célebres los casos de Newton y el árbol de manzano para el análisis de la fuerza gravitacional; de Colón observando naos en Génova o Cádiz para deducir la esfericidad terráquea antes oteada por Copérnico y Galileo; de Arquímedes limpio y ufano en los baños públicos de la antigua Roma con su *eureka* por el cálculo de un volumen cesáreo; de Darwin y su teoría expuesta en la obra sobre el origen de las especies por medio de la selección natural (Darwin, 2006); de Palamedes o los persas o los chinos por el gallardo juego del ajedrez, sólo por citar algunos casos.

Ocupémonos ahora de situaciones no tan perogrulladas, por tratarse de casos relacionados con las ciencias informáticas, enfatizando en su origen y expresando su solución sin los diamantinos formalismos matemáticos.

Problemas célebres

Cuando se estudia una determinada área del conocimiento es común tratar casos o problemas que no se han originado en un laboratorio, en un centro de

investigación o en algún entorno tecnológico o científico, sino que se han producido por algún hecho o necesidad del diario vivir. Por ejemplo, desde que el ser humano emergió en el planeta ha enfrentado el conflicto, tanto, que al ser parte natural de nuestra vida se han ideado formas de solución desde las más primitivas hasta las más elaboradas, dando origen a la teoría del conflicto social (Romero Gálvez, s.f.), un tema desarrollado en sociología, pero aplicable a la economía y al derecho. En medicina, el tratamiento de pacientes con sepsis severa de origen abdominal se puede tratar con la técnica del “abdomen abierto”, ideada en 1984 en el Hospital San Juan de Dios de Bogotá, cuando se cubrió un defecto en la pared abdominal de un paciente sometido a varias intervenciones, con una lámina plástica (polivinilo), hoy conocida en la literatura internacional como “bolsa de Bogotá” y en nuestro medio como “bolsa de Borráez” (Borráez, s.f.).

Al estudiar ingeniería, y en concreto ingeniería de sistemas y programas asociados, se presentan una serie de problemas cuyo origen parte de un caso de la vida real, como se expone en los cuatro problemas siguientes.

Acomodación de discos en las Torres de Hanoi

De niños, nuestros padres posiblemente adquirieron este juego, con el ánimo de avivar el razonamiento espacial de sus párvulos herederos. El origen del juego de las torres de Hanoi se remonta a la Europa del siglo XVIII, donde hace su aparición como un material sin duda apócrifo, en el cual se explicaba que el juego representaba una tarea que estaba realizándose en el templo de Brahma, inmerso en algún recóndito lugar del Valle del Ganges (Kruse, 1988). Según dicho material, en el momento de la creación del mundo los sacerdotes brahmanes recibieron una plataforma de bronce sobre la cual había tres agujas de diamante (origen, auxiliar y destino), según se observa en la figura 1; en la primera aguja estaban apilados sesenta y cuatro discos de oro puro, cada uno ligeramente menor que el ubicado debajo. Una versión menos exótica se distribuía en la Europa de aquel tiempo: constaba de ocho discos de cartón y tres postes de palo.

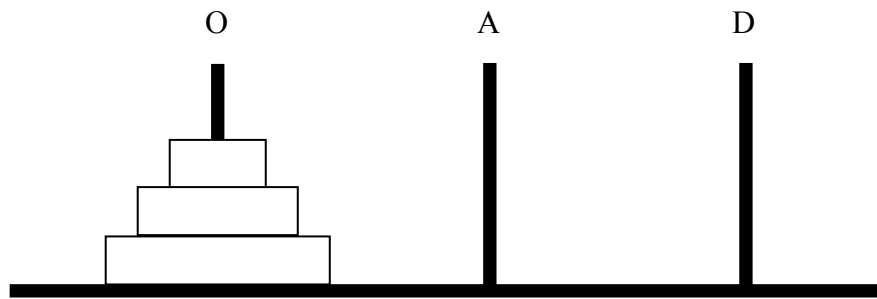


Figura 1. Discos y agujas de las torres de Hanoi, para $n = 3$

A los sacerdotes brahmanes se les encomendó la tarea de pasar todos los discos de la primera aguja a la tercera, bajo dos condiciones: sólo puede moverse un disco a la vez y ningún disco podrá ponerse encima de otro más pequeño. Se dijo a los sacerdotes que, cuando hubieran terminado de mover los sesenta y cuatro dorados discos, llegaría el fin del mundo.

La solución recursiva planteada en Kruse (1988) no se concentra en el primer paso (mover el disco de la cima hacia la aguja A o D), sino en el paso más difícil: mover el disco del fondo.

Sintetizando los pasos que se deben seguir para culminar la tarea, los sacerdotes deben proceder a:

1. Mover sesenta y tres discos desde la aguja O a la aguja A, usando la aguja D como almacenamiento intermedio.
2. Mover un disco de la aguja O a la aguja D.
3. Mover sesenta y tres discos desde la aguja A hasta la aguja D, usando la aguja O como almacenamiento intermedio.

La idea anterior se repite para los sesenta y tres discos restantes, hasta llegar al movimiento trivial de un solo disco.

Suponiendo que un sacerdote realiza el traslado de un disco a la frenética velocidad de un segundo, se requieren de $2^{64} - 1$ movimientos. Si hay cerca de 3.2×10^7 segundos en un año, la tarea total tardará unos 5×10^{11} años. Los astrónomos estiman la edad del universo en diez mil millones de años (10^{10}). Por tanto, según esta historia, el mundo durará cincuenta veces más de lo que tiene de existencia.

Recorrido por los puentes de Königsberg

Múltiples problemas en matemáticas, computación, investigación de operaciones, medicina y otras áreas, conducen a planteamientos que requieren el uso de grafos dirigidos y no dirigidos. Un grafo es una figura amorfa que consta de un conjunto de nodos o vértices conectados entre sí por un conjunto de aristas, tramos o arcos, tal como lo ilustra la figura 2, donde los vértices se representan mediante círculos y los tramos por segmentos de línea.

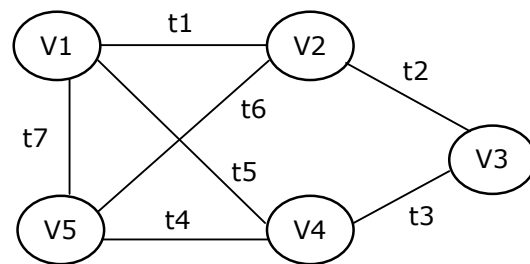


Figura 2. Grafo no dirigido

Su imaginación lo puede transportar a la variabilidad de aplicaciones de esta estructura: sistema de transporte donde los nodos son ciudades y los arcos carreteras o líneas aéreas; red de comunicación donde los vértices son servidores y las tramos flujos de datos; camión repartidor que debe visitar distintos puntos de una ciudad —vértices— para dejar su producto en determinado lapso de tiempo —tramos—; ga-

soducto donde los vértices son domicilios y los tramos la tubería; cada neurona de un ser vivo haría las veces de nodo y los mensajes entre ellas serían las aristas. De manera similar, se podría mencionar una amplia gama de casos adicionales.

El iniciador del trabajo con grafos para el modelamiento de problemas fue el matemático suizo Leo-

nard Euler, a quien en 1736 le propusieron determinar una manera de recorrer exactamente una vez cada uno de los siete puentes que atraviesa la ciudad de Königsberg (hoy Kaliningrado, Rusia), terminando en el mismo punto de partida. Los puntos se encontraban dispuestos sobre el río Pregal como se muestra en la figura 3.

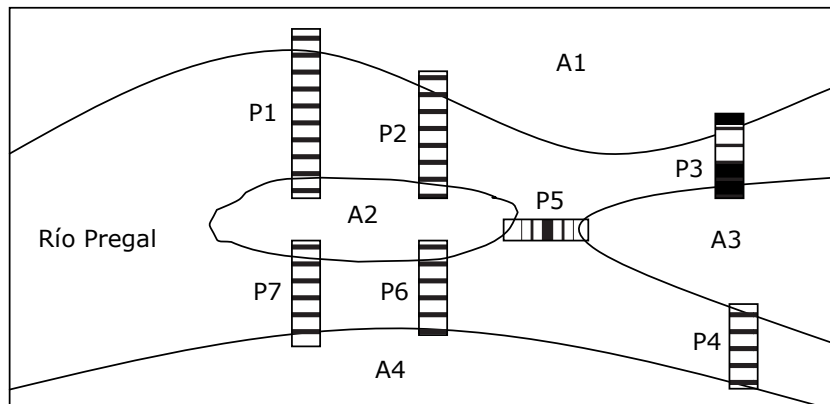


Figura 3. Problema de los puentes de Königsberg

Euler demostró que no era posible encontrar dicha ruta, fundamentado en que para cada sector de la ciudad confluye un número impar de puentes. El modelamiento de Euler consistió en un grafo, donde los sectores o áreas A1, A2, A3 y A4 son los nodos y los puentes p1 a p7 constituyen las aristas (ver figura 4).

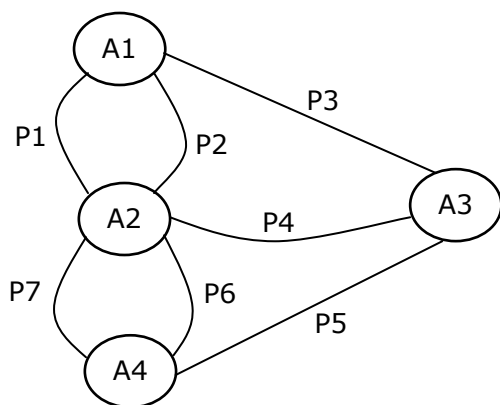


Figura 4. Modelo de Euler para el problema de los puentes de Königsberg

Penetración de sistemas informáticos en base a gusanos, virus y caballos de Troya

Gran parte del género humano siente aversión por la especie espeluznante de invertebrados llamados gusanos. Según el contexto, estos animales pueden ser benéficos o perjudiciales al hombre; *verbi gratia*, la sanguijuela es un anélido que puede causar anemia al ganado cuando no se le controla, pero en medicina es de gran utilidad para sangrías locales. Las larvas de los insectos lepidópteros causan estragos en la agricultura, no obstante en su hábitat natural forman parte de la conservación del ecosistema. Por otro lado, las picaduras de algunos de estos contráctiles seres pueden ser dolorosas, a veces letales, pero las utilidades manifiestas de la lombriz de tierra y el gusano de seda son evidentes para el bienestar de los seres humanos. Así mismo, hay gusanos planos parásitos como las duelas y las tenias que viven en el intestino del hospedero. También hay gusanos de vida libre como la acuática y carnívora planaria.

El parasitismo vegetal se manifiesta en fanerógamas, hongos, bacterias y virus. Los virus son el origen de múltiples enfermedades en el hombre (viruela, rabia, fiebre tifoidea, poliomielitis, escarlatina, tifus exantemático), en animales (brucelosis) y en plantas. Incluso virus de laboratorio son encubiertos al mundo para proyectarlos como armas biológicas.

Dejando a un lado los gusanos y parásitos, recordemos que Homero en su *Iliada* describe la epopeya de la toma de Troya por los soldados griegos escondidos en el vientre de un caballo de madera gigantesco, que los ciudadanos troyanos confundieron con una ecuestre dádiva del enemigo (Homero, 2010).

Estos aspectos biológicos e históricos se extienden al estudio de los mecanismos de seguridad y protección para los sistemas informáticos. El empleo creciente de computadores en las actividades de comercio, gobierno y milicia, reverbera a diario en grandes transacciones electrónicas de fondos, notificaciones de propiedad intelectual, datos comerciales estratégicos, registros sobre individuos e inúmeros sucesos adicionales que pueden ser revelados sin autorización o destruidos por usuarios ilegítimos.

Existen varios mecanismos mediante los cuales se puede intentar la penetración de un sistema computarizado, entre ellos, los gusanos, los virus informáticos y los caballos de Troya (Milencovic, 1994).

Un gusano informático es un programa que puede invadir las computadoras, generalmente a través de una red, y denegar servicio a los usuarios legítimos utilizando cantidades desproporcionadas de recursos de procesamiento y comunicación para su auto propagación. El nombre fue acuñado como analogía del gusano solitaria, criatura parásita que vive en los intestinos de los seres humanos y otros vertebrados infectados. Los primeros gusanos informáticos eran benignos, pues fueron diseñados como mecanismos para extender grandes computaciones a través de una red de computadores, con el fin de hacer uso del tiempo de máquina inactivo. Sin embargo, como sus homónimos, los gusanos informáticos pueden convertirse en parásitos cuando la especie prolifera.

Son bien conocidos ciertos casos de gusanos artificiales que han conseguido extenderse a través de redes públicas, llegando a infectar a miles de computadores *ipso facto*. La línea habitual de ataque son las listas de correo de usuarios que contienen los nombres y di-

recciones de otras máquinas alcanzables; tras obtener acceso a una lista de correo, el gusano envía copias de sí mismo a algunas o todas las máquinas listadas, replicando el proceso de propagación sobre las copias recientes. La infección se extiende con rapidez y ahoga el sistema entero al consumir tiempo de proceso y ancho de banda de la red, de tal modo que se impide la realización de todo trabajo o comunicación.

Las salvaguardias frente a los gusanos informáticos están dadas por programas que refuerzan la seguridad del sistema, especialmente en las áreas de utilidades de correo, contraseñas y control de acceso a las listas, y por puntos de comprobación en el sistema de comunicación que eviten la propagación de los gusanos.

Los virus informáticos son trozos de código que infectan otros programas, generando eliminación de archivos o corrupción del bloque de arranque de un disco. Un virus informático, como su homónimo biológico, no puede funcionar por sí mismo, sino que debe introducirse en un programa anfitrión. Tras infectar una máquina o programa, puede permanecer aletargado durante un lapso de tiempo para evitar sospechas. La infección se extiende habitualmente a través de la adquisición de programas obtenidos directa o indirectamente de redes informáticas y boletines electrónicos en forma de programas de dominio público, aunque se tiene noticia de casos en los cuales se han adquirido programas corrompidos a través de software comercial legítimo, como resultado de la contaminación de algunos distribuidores y vendedores de software.

A diferencia de los gusanos, los virus no se propagan por sí mismos a través de las redes; por lo general se extienden del programa portador infectado que se ejecuta en diversas máquinas. La propagación de virus es posible en cualquier computador programable, sobre todo en computadoras personales con sistemas operativos desprovistos de protección debido su instalación ilegal. No hay una vacuna general conocida para los virus informáticos. Las medidas preventivas comunes incluyen el filtrado preventivo de todo software de reciente adquisición, las copias de respaldo frecuentes y una combinación de utilidades, tales como comprobadores de integridad, programas de vigilancia y supresores de virus.

Un programa caballo de Troya es aquel que puede ocultar intencionalmente parte de su funcionalidad,

con frecuencia dañina, buscando transmitir datos o derechos de acceso del usuario a otro que hace las veces de intruso (Milencovic, 1994). Un sencillo programa caballo de Troya se puede usar para robar contraseñas de usuario imitando al programa legítimo de apertura de sesión y reproduciendo fielmente la secuencia y diálogos de presentación normal. Estos programas son fáciles de implantar en sistemas cuyas terminales se hallan en recintos públicos, dejando una copia activa en un terminal y haciendo que simule las pantallas de presentación y despedida. Otras versiones sofisticadas de esta tipología de programas anulan totalmente la utilidad que están suplantando.

Orden en la mesa de los filósofos comensales

Las computadoras modernas pueden ejecutar varias tareas al mismo tiempo, lo que da origen en el ámbito de los sistemas operativos al término multitarea. Si reemplazamos el vocablo tarea por proceso, surge el término multiproceso, y si consideramos un proceso como un programa de computadora en ejecución, entonces multitarea se convierte en multiprogramación.

La multiprogramación realmente no se presenta, porque, en rigor, en cualquier instante de tiempo la unidad central de procesamiento (CPU) está ejecutando sólo una tarea. La ilusión de paralelismo se presenta por la rapidez de ejecución de las instrucciones.

Un caso patético de paralelismo se presenta cuando se espera la apertura de una página web desde internet y se decide abrir un archivo de hoja electrónica, para ser impreso de inmediato; en éste y otros casos, el sistema operativo se verá obligado a tramitar una comunicación entre procesos y a solucionar todos los impasses que se puedan presentar para finiquitar con éxito todas las tareas pendientes.

La literatura sobre sistemas operativos está repleta de interesantes problemas clásicos de comunicación entre procesos, uno de ellos conocido como el problema de los filósofos comensales (Tanenbaum, 1988). El planteamiento de la situación data de 1965, cuando Edsger Dijkstra formula y soluciona el problema

de manera sencilla y peculiar, imaginando a cinco filósofos sentados en torno a una mesa circular, dispuestos a ingerir cinco platos de delicioso espagueti singularmente resbaladizo, tanto, que un filósofo necesita de dos tenedores para poder comer. Cada filósofo debe controlar su voraz apetito pues sólo se cuenta con igual número de tenedores ubicados entre cada plato, tal como lo ilustra la figura 5.

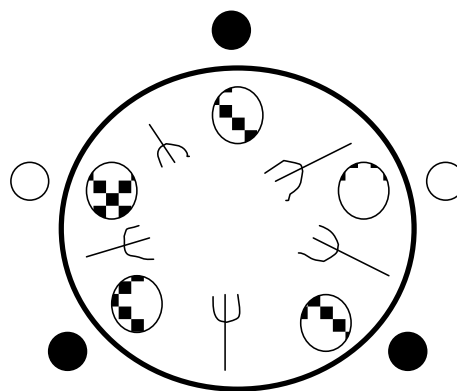


Figura 5. Vista superior del quintuplo de platos, tenedores y filósofos, dos de ellos rapados o posiblemente calvos

El problema conduce a imaginar comensales italianos, pero también se puede pensar en matemáticos árabes comiendo kebab, o vegetarianos hindúes consumiendo Maha Prasada, o místicos orientales disfrutando el arroz chino —aquí los tenedores se convierten en palitos—, o alguna otra situación gastronómica casual del avatar terreno.

Prosiguiendo con el problema, la vida de un filósofo consta de periodos alternos de comer y pensar, supuesto en tanto insulso pero necesario en el esquema. Cuando alguien siente hambre, intenta asir dos tenedores, el de su izquierda y el de su derecha, en cualquier orden. Si logra tomar ambos tenedores, engulle unos bocados, baja los cubiertos y sigue filosofando.

La solución obvia e inmediata para un número N de filósofos, tomando la variable i como el i -ésimo comensal, es la siguiente:

```

problema_filósofos (N) {
    repita para cada filósofo i = 1, 2, ..., N
        filósofo (i)
        fin_ciclo_repita
    }

    filósofo(i) {
        repita mientras exista apetito {
            El filósofo i está meditando
            Tomar tenedor de la izquierda
            Tomar tenedor de la derecha
            Disfrutar del espagueti
            Colocar tenedor izquierdo sobre la mesa
            Colocar tenedor derecho sobre la mesa
        }
    }
}
    
```

La solución anterior por lo obvia es equivocada: si los cinco filósofos toman los tenedores a la vez, ninguno de ellos podrá tomar el tenedor derecho y habrá estancamiento.

Conclusión

El estudio de la ingeniería de sistemas, la ingeniería en software, la ingeniería informática y demás titulaciones relacionadas, así como otras áreas del conocimiento, conllevan problemas que planteados desde su origen cotidiano, hacen del estudio algo más agradable, con menos traumatismo cognitivo, para conducir en conjunto a formalismos tecnológicos y científicos que cualifican el proceso formativo del futuro profesional.

Referencias

Borráz, Oswaldo (s.f.) Abdomen abierto. Utilización del polivinilo. *Revista de Cirugía. Estudios analíticos*. [documento en html]. Consultado el 3 de junio de 2011 en: <http://www.encolombia.com/medicina/cirugia/cirugia16101-abdomen.htm>

Darwin, Charles (2006). *El origen de las especies*. Prólogo de Faustino Cordón. Madrid: Editorial EDAF.

Homero. (2010). *La Iliada. Clásicos de Grecia y Roma*. Madrid: Alianza Editorial.

Kruse, Robert L. (1988). *Estructuras de datos y diseño de programas*. Naucalpán de Juárez, Prentice-Hall Hispanoamericana.

Milencovic, Milan. (1994). *Sistemas operativos*. Madrid, McGraw-Hill.

Romero Gálvez, Antonio (s.f.). *Teoría del conflicto social*. Consultado el 3 de junio de 2011 en: <http://www.gestiopolis.com/recursos4/docs/ger/tene-gouno.htm>

Tanenbaum, Andrew S. (1988). *Sistemas operativos: diseño e implementación*. Prentice-Hall Hispanoamericana, Naucalpan de Juárez, México.

Villalobos, Jorge A. (1996). *Diseño y manejo de estructuras de datos en C*. Santafé de Bogotá, McGraw-Hill.